Procedure for sending diagnostic data to Pulsar via the Transition Module

SH 10/26/2005
Edited 4/5/06

The current design that is loaded into the L2-Pulsar FPGA will send the pattern that is loaded into it's Diagnostic RAM to the Pulsar every time the FPGA receives a L1 Accept signal from the backplane. The pattern that is sent follows CDF note 7772. The driver loop routine requires a halt signal to go low and then back to high(this requirement can be overwritten by accesing the control register of the L2_Pulsar FPGA and setting the correct bit) followed by a Level 1 Accept to invoke the Driver routine.

The procedure for setting up the routine is such:

1. Enable the Diagnostic Mode bit for the L2-Pulsar FPGA in the boards Control register.
2. Load the Diagnostic RAM of the FPGA with the appropriate file.
3. Enable the TX_Mezzanine board by accesing the TX_Mezzanine register in the FPGA.
4. Enable the Driver Loop command in the FPGA control register.
5. Invoke the testclock to send a Halt signal to low and then back to high followed by a L1_Accept onto the backplane.

The SXFT_V2 menu driven software that Rod Klein has written provides a simple method for performing the five steps above. A text file has been made that follows CDF 7772 data format and is included at the end of this note.

The Transition module and TX_Mezzanine board needs to be plugged in directly behind the Finder board that is being exercised. The TX board provides the clock for the FPGA to run it's state machine and synchronized the data to the same clock that the SERDES is operating with.

Follow the below sequence to prepare the board for sending data to the Pulsar.

Invoke SXFT_V2 software. The software will ask for Crate Controller name and slot locations for modules. The first menu will be the "Main Menu":
Select option 1 – "Register Menu"
Select 2(Control/Status) from menu.
Toggle bit 14 to 1 > Pulsar Driver Diagnostic Loop Mode
Select Q from menu.
Select option 14 – "L2 – Driver Internal Status and Control"
Select option 7 – "L2 Output RAM"
Select option 6 – "Load output RAM from File"
Input the filename – "L2_pulsar_data_walk1.txt"
The data from the file is now loaded into the diagnostic RAM in the L2-Pulsar FPGA.
Select Q from menu.
Select 5(TX- Mezzanine Register)
Toggle bit 4 to 1 > TX Mezanine Enable
Select Q from menu.
Select 2(Status/Control Register)

Toggle bit 7 to 1 > Disable HRR requirement
Toggle bit 2 to 1 > Driver Loop enable

The Finder will now begin shifting up to 256 slices of data out of the L2-Pulsar path to the backplane every time a L1 Accept is recognized by the Finder board. The number of slices is dependent upon which one of the slices contains the bits that inform the system that this slice is the "Trailer" word(bits 14 and 15 are '1'). The transmission of data will stop when a Trailer word is found.

To stop the sending:
Select option 2 – "Toggle Driver Loop Enable" - - bit should be off in the menu.

You can check the L2-buffers in the FPGA by selecting
10, 11, 12, or 13. The format for the L2-buffers is a 32 bit word of data that is sent on path A. So the even words are in the 16 LSB bits and the odd words are in the 16MSB. The 2 LSB of the second word represent the Buffer # and will select which of the 4 L2-buffers the data is stored into.

The Diagnostic RAM is a 32 bit device and the L2-Pulsar provides two 16 bit data paths to the Transition module. The file below loads both paths up with the same data.

** (MSB...LSB) **
** bits 0-15 are output on port A ** and also stored in the L2-buffer
** bits 31-16 are output on port B **
** review CDF note 7772 for explanationof bits 0-15 **
** The subcell Number is decoded into the subcell bits **
** there are 90 slices of data - becuse the dedcoding of **
** the numbers 1-36 into a hex number produce 90 bits **
aaa1aaa1 ** Header word 1 **
555c555c ** Header word 2 ** ! the two LSB are 00 = buffer # 0
00820082 ** subcell 1 & 2 **
02060206 ** subcell 3 & 4 **
030a030a ** subcell 5 & 6 **
040e040e ** subcell 7 & 8 **
05120512 ** subcell 9 & 10 **
06160616 ** subcell 11 & 12 **
071a071a ** subcell 13 & 14 **
081e081e ** subcell 15 & 16 **
09220922 ** subcell 17 & 18 **
0a260a26 ** subcell 19 & 20 **
0b2a0b2a ** subcell 21 & 22 **
0c2e0c2e ** subcell 23 & 24 **
0d320d32 ** subcell 25 & 26 **
0e360e36 ** subcell 27 & 28 **
0f3a0f3a ** subcell 29 & 30 **
103e103e ** subcell 31 & 32 **
11421142 ** subcell 33 & 34 **
12461246 ** subcell 35 & 36 **
00020002
00040004

00060006
00080008
000a000a
000c000c
000e000e
00100010
00120012
00140014
00160016
00180018
001a001a
001c001c
001e001e
00200020
00220022
00240024
00260026
00280028
002a002a
002c002c
002e002e
00400040
00420042
00440044
00460046
00480048
004a004a
004c004c
004e004e
00800080
00820082
00840084
00860086
00880088
008a008a
008c008c
008e008e
01000100
01020102
01040104
01060106
01080108
010a010a
010c010c
010e010e
02000200
02020202
02040204
02060206
02080208

020a020a
020c020c
020e020e
04000400
04020402
04040404
04060406
04080408
040a040a
040c040c
040e040e
08000800
08020802
08040804
08060806
08080808
080a080a
080c080c
080e080e
10001000
10021002
10041004
10061006
10081008
100a100a
100c100c
100e100e
20002000
20022002
20042004
20062006
20082008
200a200a
200c200c
200e200e
40004000
40024002
40044004
8003C003 ** Trailer **